

# Registered Motion of an Ultrasound Transducer with a 6 Degrees of Freedom Industrial Robot for the Construction of Tissue Displacement Maps

SENIOR HONORS THESIS | BIOMEDICAL ENGINEERING  
UNIVERSITY OF NORTH CAROLINA AT CHAPEL HILL

STEPHEN A LEE

Approved:

---

Gianmarco Pinton | Thesis Advisor

Devin Hubbard | Reader

Robert Dennis | Reader

## **Abstract**

In the past, brain injuries were modeled using head acceleration. However, this is only an indirect relation to the brain tissue. We aim to measure brain motion more directly and accurately. New innovation in ultrasound scanning is a vital key to producing accurate models. In this study, we aim to accurately position an ultrasound transducer with six degrees of freedom ABB IRB 120 robot arm. The overall goal of this method is to produce 3D vectorial maps of brain displacement through time (ie 7 dimensions total). This will allow us to characterize the propagation of shear shock waves in the brain and to establish a link to brain injuries.

## **I. Background**

The Centers for Disease Control and Prevention (CDC) estimates that 1.7 million Americans will be hospitalized annually as a result of Traumatic Brain Injury (TBI). TBI also contributes to a third of all injury-related deaths in America [1]. Many more injuries go unreported and at least 80,000 individuals will develop a disability due to TBI related injuries [2]. As such, TBI represents a major population of deaths which can be dramatically decreased and prevented as a result of improved research models. Superior models not only increase the overall understanding of the brain, but will also spur innovation in Traumatic Brain Injury prevention and treatment.

A TBI is produced by a violent movement of the head [3]. Shear waves produced from the initial impact of the skull transfers to the soft brain tissue and because brain tissue is gelatinous and pliable, tears and lesions in seemingly random locations in the brain are produced. Though the head (skull) can be monitored using accelerometers and

other tools, the brain cannot because until now, no imaging tools allow detection of brain motion. However, high frame-rate (10,000 images / second) ultrasound transducers makes imaging brain motion possible.

The study of shockwave propagation in elastic materials such as the brain is a new field that can greatly benefit research in Traumatic Brain Injury. Currently, direct experimental observation of shockwave propagation through soft solids is done using an ultrasound transducer, an Agar-gelatin phantom, and a vibrator [4]. Direct observation is made possible by high frame-rate ultrasound scanners that have the capability of producing images of up to 10,000 frames/second and radio-frequency (RF) motion tracking algorithms [5].

Ultrasound transducers produce short bursts of ultrasound frequency which can be measured by a hydrophone. The hydrophone, coupled with a pre-amplifier, can record acoustic waves in water and change them into electrical energy [6]. The pulse can then be recorded, filtered, and analyzed in a computer. Using the ultrasound transducer and the hydrophone, the intensity of the ultrasound pulse can be plotted by position.

According to a survey produced by World Robotics in 2010, six-axis robot arms have traditionally been used in factory, machining, and line assembly tasks [7][11]. But some studies have shown that accuracy is reliable enough for research use. To coherently form an ultrasound image, wavelength and speed of sound is needed for positioning accuracy. For articulated robots such as the ABB IRB-120, the repeatability error increases with reach distance. The larger the reach distance, the lower the repeatability will be, generating more error and less accurate models and images. Table 1 shows data for 3

selected ABB IRB robots [8]. However, if reach distance can be reduced, the error of repeatability can be drastically decreased. Articulated robots, such as the ABB IRB 120, are also cost-effective, flexible and have a large working envelope.

*Table 1 - Robot reach distance and repeatability*

ROBOT MODEL	REACH DISTANCE (MM)	REPEATABILITY (MM)
<b>ABB IRB 120</b>	580	$\pm 0.01$
<b>ABB IRB 140</b>	810	$\pm 0.03$
<b>ABB IRB 1410</b>	1440	$\pm 0.05$

As reach distance increases, the rate of error also increases. Smaller articulated robots will have much greater accuracy and precision as long as the task does not lie beyond the scope of normal operation. Limiting factors also include the area of the attachment on the arm. Large attachments will limit the way the robot can move and rotate, essentially limiting the amount of space the robot will be able to reach. There are no commercially available attachments for ultrasounds transducers. Therefore, customized and unique attachments are required for use in research.

The ABB IRB 120 can be easily programmed using RobotStudio [ABB Robotics, Zürich, Switzerland] to define targets and paths for simple tasks and operations. By defining the orientation and targets of the robot in virtual space, the data can be synchronized with the real controller and controlled in real time. Also simultaneously, when defining points and targets, RobotStudio generates RAPID code that can be modified just like a text editor. The RAPID code is the basis of handshakes between the robot

controller and PC SDK (Software Development Kit) and the SDK is used to operate the robot controller without the use of manual programming in RobotStudio. Files can be transferred from the computer directly to the robot controller, which will be executed instantly after input from the user. In this study, the input was from the ultrasound transducer itself. Therefore, it is necessary to produce programming for communication between the ultrasound transducer, the Linux terminal and its partner, and the robot controller.

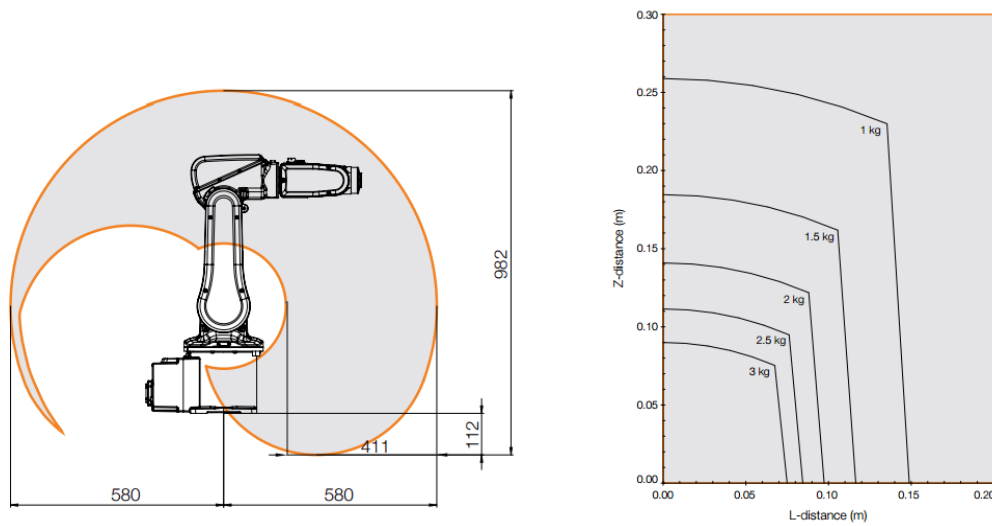
Linear motion in RAPID is carried out using Euler rectangular coordinates [9]. Rotational movement is defined by the use of quaternions. Quaternions are used mainly in robotics and aerospace engineering as an alternative to traditional angles. Though Euler angles are simple and intuitive, they are limited by a phenomenon called “Gimbal Lock” [10]. When Euler angles approach  $\pm 90$  degrees, there is no method to measure orientation. However, quaternions can be represented as a vector with four components; a scalar plus three imaginary components giving the orientation no matter what angle.

## **II. Aims**

The objective of this study is to accurately position the ultrasound transducer using an ABB IRB 120 robot with six degrees of freedom. The system positions the ultrasound transducer with 10 microns of precision to obtain registered data sets of tissue displacement. Additionally, the technique in this study will enable further research into creating 7D models. The methods I have developed involve two manufacturing processes, one programming task, and one data registration technique.

## Manufacturing

To accurately attach the ultrasound transducer with the ABB IRB-120 robot, an attachment had to be machined. For success in attaching the ultrasound to the robot, the holder must be water proof and has to clamp the transducer in the same position ( $\pm 10 \mu\text{m}$ ) every time. Additionally, the holder must also be detachable and break down into washable pieces after each experiment due to exposure to brain tissue. Because the effective range of the robot arm decreases depending on how large the attachment is and how close it is to the base of the arm, the material needs to be lightweight, effectively decreasing the error of repeatability and reducing the stress on the robot arm. Rotating the transducer becomes increasingly difficult if the attachment limits where the arm can rotate in space as shown in figure 1.



[9]

Figure 1 – Illustration of the effective range of ABB IRB 120 robot arm under different distances and stress

Previous studies have created 4 dimensional tissue displacement maps using a standard plastic bin capable of holding water with an open top to allow for 1 dimensional scanning and a motorized component with three degrees of freedom [4][5]. However, 7 dimensional maps will allow reconstruction of the tissue displacement in 3 more dimensions giving more information to understand how brains react in traumatic injuries. To construct 7D maps, a box containing thin and clear windows must be manufactured. The windows must be significantly thin to minimize reflection artifacts that degrade ultrasound images. Additionally, the windows should also be flush with the inside wall to minimize boundary effects when producing shockwaves with a vibrational apparatus. Furthermore, the box must be waterproof and small enough to fit into a vacuum chamber for degassing water, gelatin, and ex-vivo soft tissue.

#### **A. The Clamp**

A Verasonics ultrasound system (Verasonics Inc., Redmond, WA) was coupled with a Philips L7-4 linear array transducer (Philips Healthcare, Andover, MA) in order to capture physical behavior not observable at normal frame-rates. The clamping method was determined by the unique shape of the L7-4 ultrasound transducer and the ease and accessibility of materials.

Both faces of the transducer have three indents that can be used to hold the transducer in place [figure 2]. The sides of the transducer also uses a unique curve which was intentionally made to complement the human hand. The neck near the head of the transducer is much wider than the tail end of the transducer. Due to these unique features, molding becomes an optimal and viable choice.



Figure 2 – Picture of the Ultrasound Transducer side divots and curved handle

The ultrasound transducer will be frequently subjected to liquid and soft tissue therefore the molding material must be liquid-resistant and able to be disinfected. Two-part aluminum-filled liquid epoxy [Devcon, Danvers, MA] was chosen because of it’s stability, hold, and ability to be machined [figure 3]. The most important characteristic of the epoxy mold is the repeatability of accurate placement of the transducer despite temperature changes. Because the transducer will be subjected to room temperature fluctuations, the epoxy mold must not yield and deform or it will jeopardize the integrity of the holder. The two-part epoxy was mixed according to weight and the ratio specified below. Microns per Celcius was calculated to show that with a range of 20 to 24 degrees Celsius (room temperature) the molding material will only expand 9.144 microns, less than robots repeatability.

Table 2– List of physical properties of the Aluminum-filled epoxy resin.

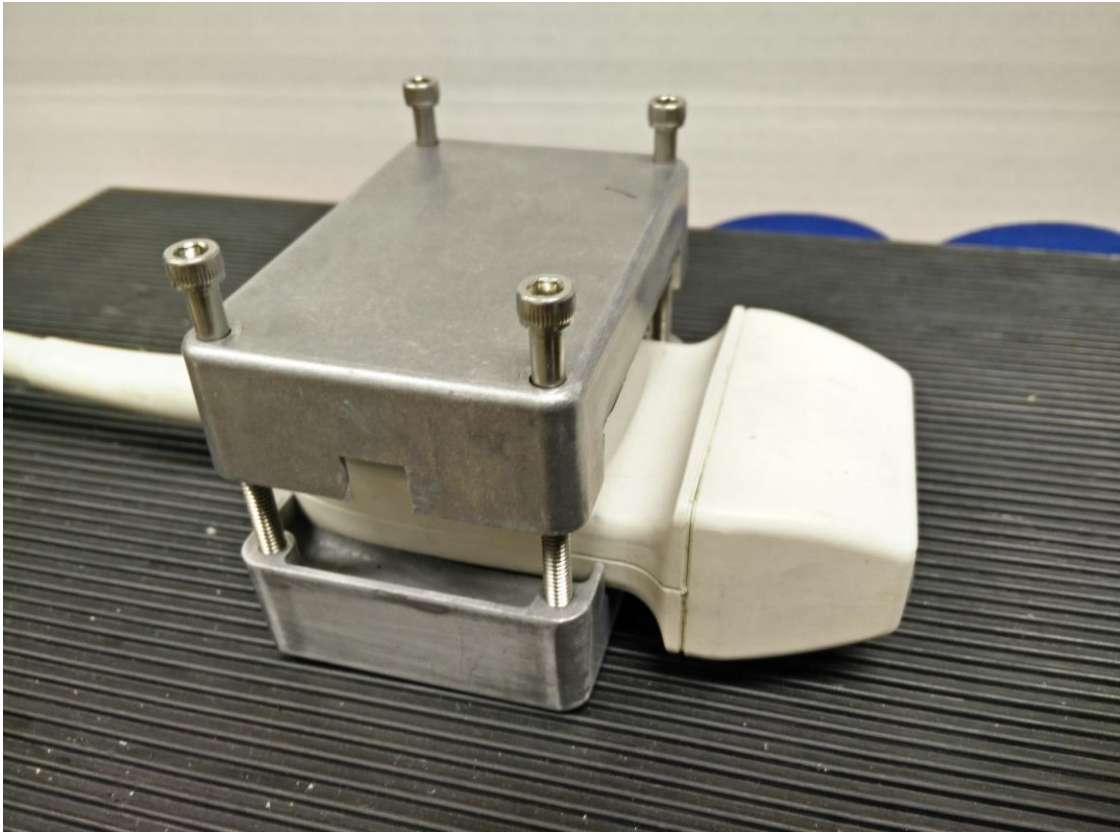
<b>COEFFICIENT OF THERMAL EXPANSION</b>	50E-6 IN/°F	$50 \times 10^{-6} \frac{\text{in.}}{\text{Fahrenheit}} \times 0.0254 \frac{\text{m}}{\text{in.}}$ $= 1.27 \frac{\text{microns}}{\text{Fahrenheit}} = 2.286 \frac{\text{microns}}{\text{Celcius}}$
<b>FUNCTIONAL CURE</b>	16 hours	
<b>MIX RATIO BY WEIGHT</b>	9:1	



To create an accurate mold of the transducer without compromising the transducer itself, a release agent was used. In this study we used a water-soluble liquid mold release [Evercoat, Cincinnati, Ohio]. In addition to the releasing properties, the mold release only requires one or two full coats and 30 minutes to settle.

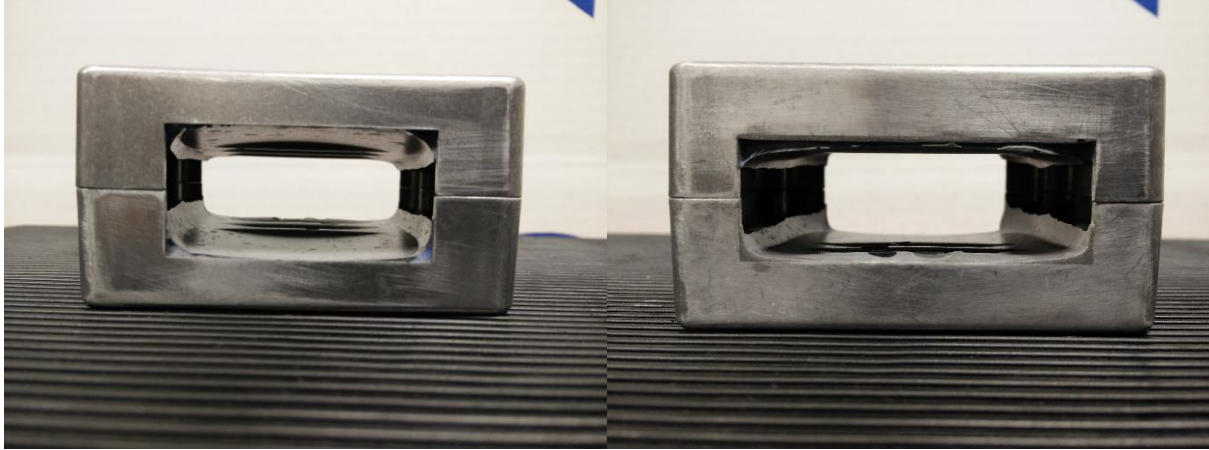
The selected encasing for the mold was two Hammond 1550p aluminum encasings without the lid [Hammond Manufacturing Co. Ltd, Guelph, Ontario, Canada]. The 1550p aluminum casings were originally used for electronic storage which provides an explanation for its smaller size. Additionally, aluminum material is relatively easy to mill and modify. The size outlines the transducer well and does not add too much bulk to the overall attachment. Furthermore, holes were already cut into the box providing easier alignment of the two halves. There is extra material around the screw holes to drill larger screws.

There were three components to the robot transducer attachment: the box encasing, the aluminum molding, and the L-bracket which the box will sit upon and create a 90 degree angle to the robot arm face. The two 1550p aluminum encasing boxes became the clamp that holds the mold of the transducer. Each of the screw holes were drilled out in preparation for being screwed together. One half of the box encasing was tapped and the other was drilled straight through. This allows needing only one screw to go through both aluminum encasings and sequentially hold the transducer in place. Screws proceed through the L-bracket, the untapped box, and finally the tapped box enclosing the transducer as seen in figure 4.



*Figure 3 – Picture of the ultrasound transducer between the two halves of molded clamps. All of these pieces are held together by 10-24 hex screws.*

Eight holes were drilled into the already existing holes of both Hammond 1550p boxes since the pre-existing holes do not penetrate all the way through. The drill bit was selected to house a 10-24 screw. One of the boxes was selected as the top and all the holes were tapped by hand. Since the height of the transducer is 20 mm and the combined height of the two boxes together is 42mm, 5 mm were milled off of each box. The milling process was done manually rather than using a preset program enabling better control. The boxes were fitted together properly with a screw, square cutouts of the side for the transducer can now be made as shown in figure 5. These square cutouts are the size of the transducer tail and neck.

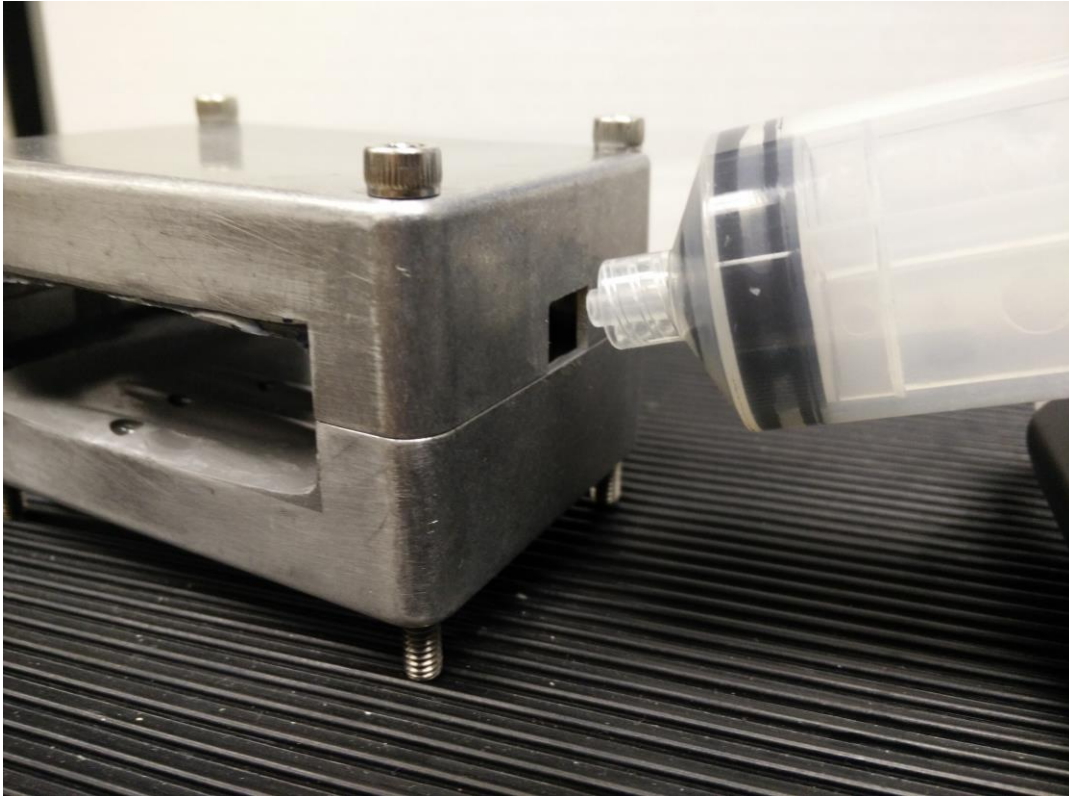


*Figure 4 – Picture of the back and front square cut outs for the butt and head of the ultrasound transducer*

Due to the sloped handle on the transducer, both cutouts needed to have different heights and widths. The cutouts were made using a programmed frame on the milling machine and the sizes of the cutouts are displayed in Table 2. Now that the main body is finished, two 5mm x 5mm squares were cut out from one half of the box. The holes were cut on the uncut sides to allow access to the interior for mold injection [Figure 6]. After the milling process, edges can be taken off using fine-grit sandpaper in order to prevent damage to the robot and the transducer head.

*Table 3 – Square cut-out dimensions*

<b>CUTOUT</b>	<b>HEIGHT (MM)</b>	<b>WIDTH (MM)</b>
<b>FRONT SQUARE</b>	18.83	55.03
<b>BACK SQUARE</b>	22.80	46.74

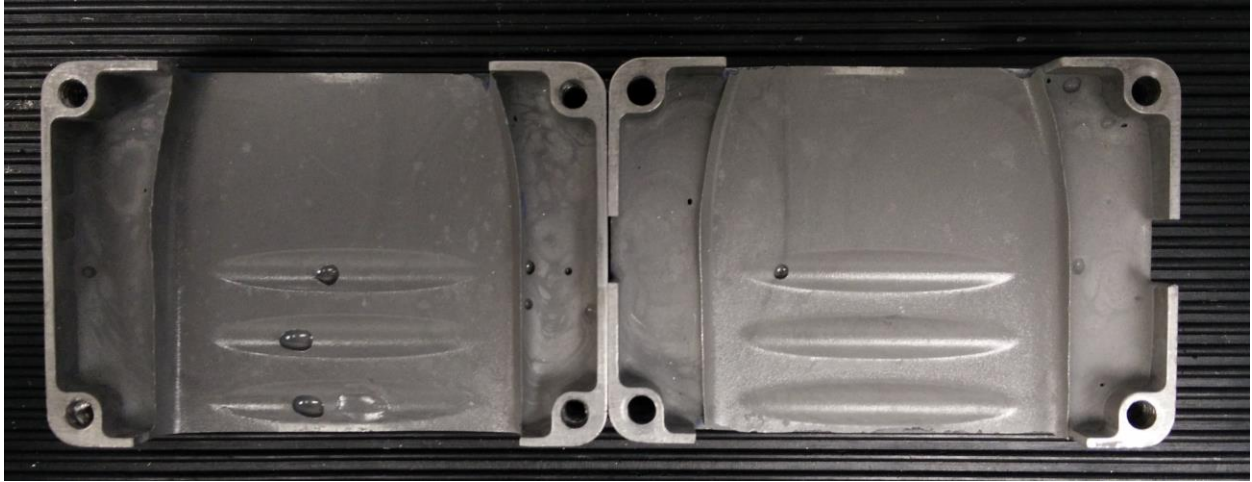


*Figure 5 – Side square cutout for injection of aluminum-filled epoxy resin via syringe*

The two-part aluminum filled epoxy was carefully mixed at a 9:1 resin to hardener ratio by weight on a balance scale. Approximately 7 oz. or 200g of epoxy resin (100g for first half mold and 100g for second half) was mixed slowly in order to minimize the number of air bubbles for 30 minutes. The epoxy was then inserted into large 100 ml syringes.

Before molding, the releasing agent was applied. Soft-bristled brushes were used to apply an initial coat of the release agent and after 30 minutes, another coat was applied. After drying, we placed the ultrasound transducer inside the box frame, screwed it in with four long 10-24 hex screws and further clamped it down. Because the transducer is not perfectly straight, soft molding clay was applied to cover any holes between the transducer and the box frame at the cutouts. We then slowly injected the lower half of the box with

100g of epoxy resin as to decrease air pockets when curing. The mold was left to cure for 48 hours before release. Steps were then repeated to produce the second mold.



*Figure 6 – Picture of Aluminum molding after curing. Three divots are imprinted in each mold and the unique curve of the transducer is outlined.*

#### **A. The L-bracket**

Two different L-brackets were made with different orientations of the transducer and different lengths away from the robot arm. The different lengths allow for different positioning of the robot base relative to the box containing the gelatin phantom.



*Figure 7 – Ultrasound transducer orientation mounted on the ABB IRB 120 Robot arm*

The initial L-bracket was milled to have the transducer point straight out of the base of the robot arm [figure 8] and 60mm off center to ensure the safety of the electronic wiring tail of the transducer. 4 M5 holes were cut in one side of the L bracket corresponding to the robot hand and 4 10-32 sized holes were cut according to the holes in the 1550p displayed in Figure 10. Extraneous material (represented by white diagonals) was then shaved off using a mill to increase the rotational leeway of the robot [Figure 9].

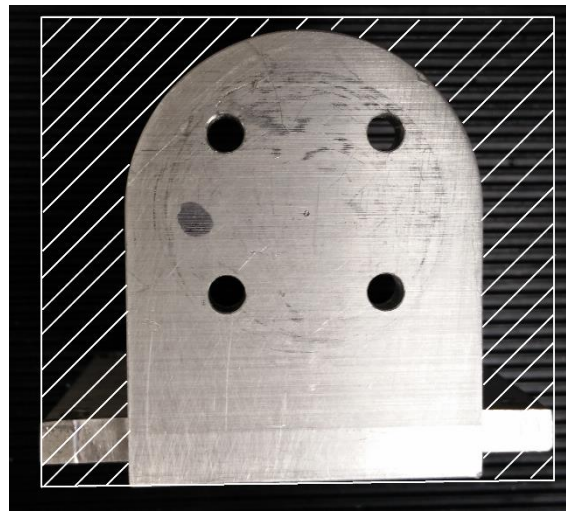


Figure 8 – Resultant L-bracket after extraneous material (white diagonals) was milled away

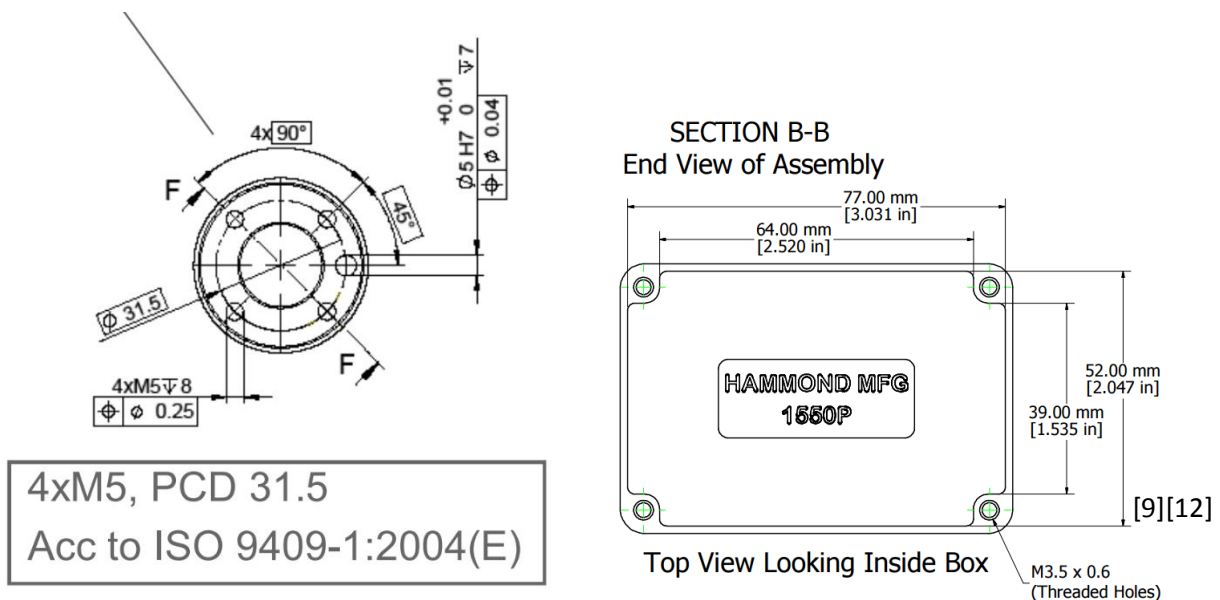
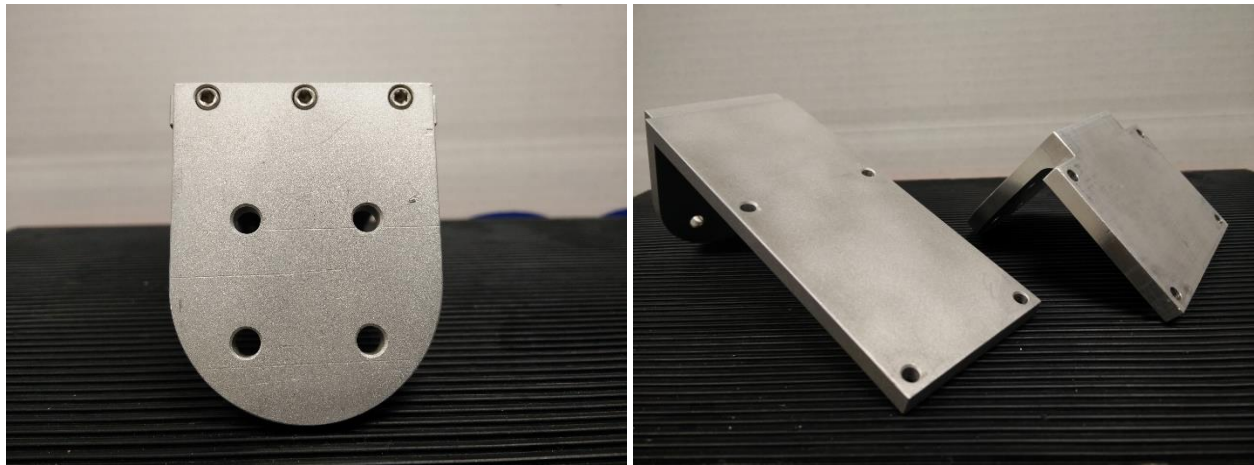


Figure 9 – Robot screw positioning and the aluminum 1550p encasing screw positioning



The second L-bracket was made to submerge the ultrasound transducer without having the robot arm subjected to water. The distance away from the base of the robot arm is 30 mm longer than the shorter L-bracket. This attachment was made by milling two different rectangular pieces and then joining them at the edges using three hex screws [Figure 11]. The orientation of the transducer is now orthogonal to the base of robot arm and offset by 53.23 mm in the x direction, 71.04 mm in the y direction, and 97.04 mm in the z direction.

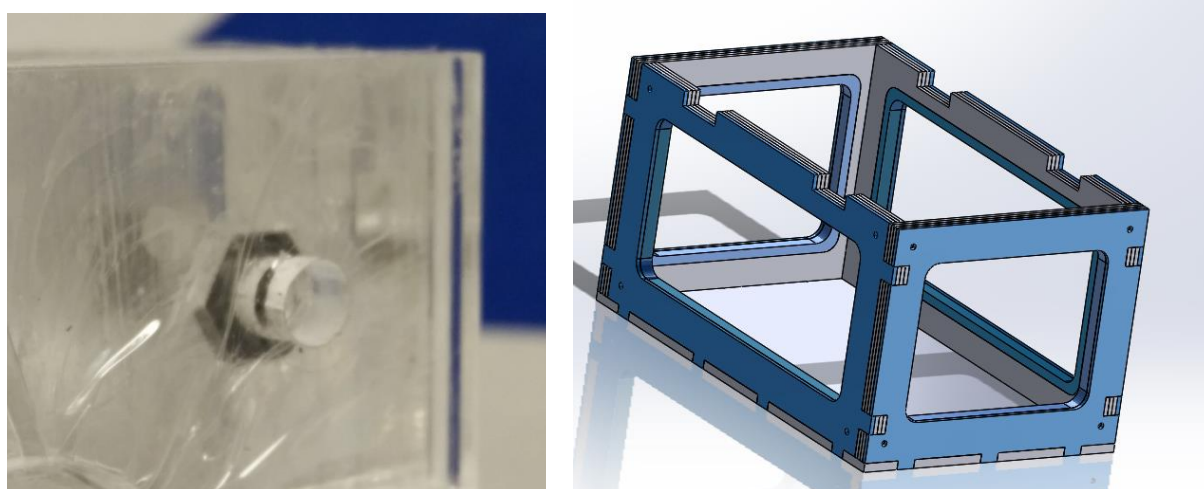


*Figure 10 – Picture of the three hex screws that conjoin the two pieces of the longer attachment and a picture of both the shorter attachment with the orientation of the transducer straight out of the robot hand and the longer attachment with the orientation of the transducer orthogonal to the robot hand.*

### **C. The Box**

The box was designed using 3D CAD design software [Solidworks Corporation, Waltham, MA] and the material for the frame was clear plastic acrylic. The design called for .118", .237" and .354" thick acrylic [Delvies Plastic Inc., Salt Lake City, UT]. For the window, .002" clear Mylar sheets were used and cut to fit in the frame [Tap Plastics Inc., San Leandro, CA]. Similar to how a drum is tightened or how window screens are made, the Mylar sheet is pressed into the frame and tension is equally distributed by tightening the

screws. A capture-nut technique was employed to screw on the window frame [figure 12] giving the user the ability to tighten loose parts of the Mylar. Three .118" acrylic plates were laser cut as shown in Figure 12 to create the inner box frames with the middle sheet containing the M4 nuts. The outer plate is also .118" acrylic and can be screwed on. Methylene Chloride was used to bond acrylic together and silicone caulk was used to waterproof the box.



*Figure 11 - Capture Nut technique in the acrylic frame and SolidWorks rendering of laser cut box frame*

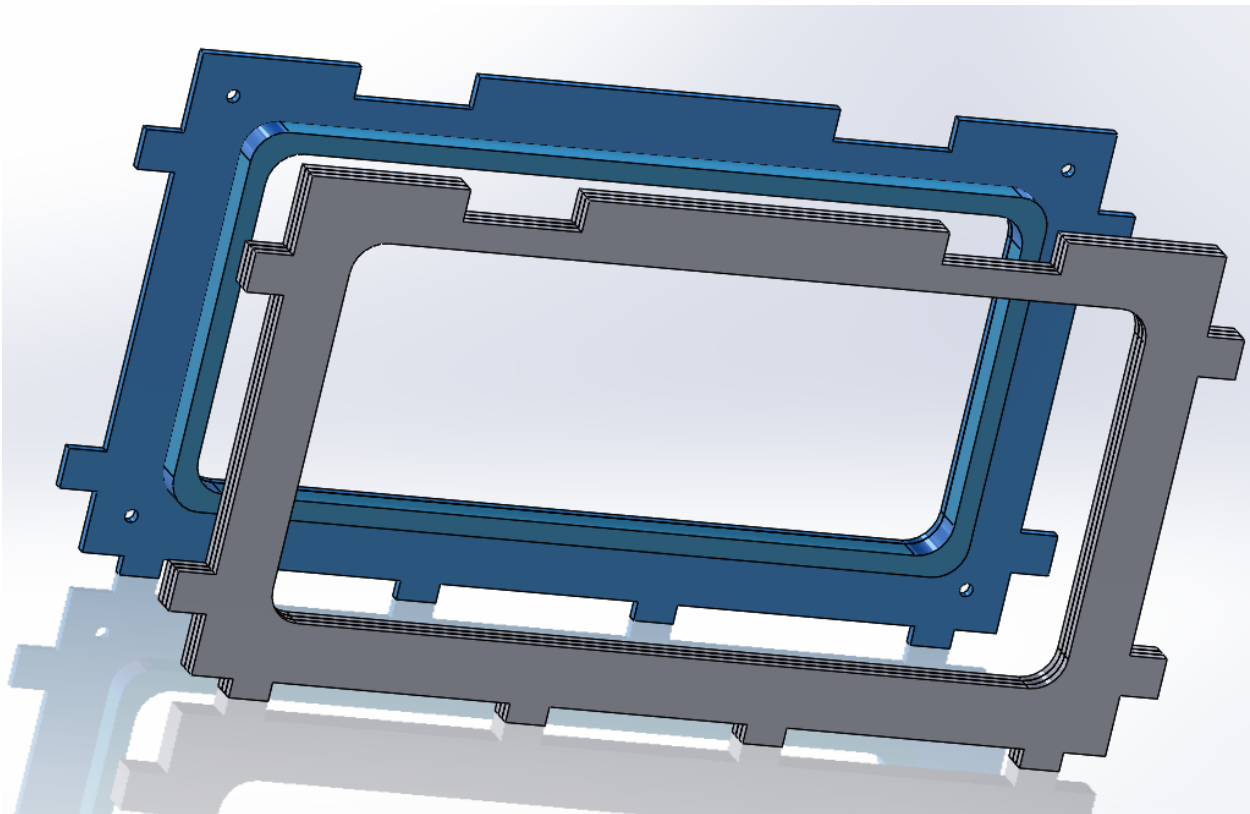
All four sides utilized the same techniques and methods of construction. The SolidWorks files were converted to a form that the laser cutter can recognize and each plate was cut according to the blueprints. Rough edges were sanded down with fine-grit sandpaper. Furthermore, the base was cut with divots to increase the surface area of contact in preparation for the dichloromethane.

The inner plate was cut without holes, the next adjacent plate was cut with hexagonal holes to cradle the hex nuts, and the next plate was cut with a hole centered with the hexagonal holes to “capture” the nut and allow for a screw to pass through. All three of



these layers were glued together with Dichloromethane using a 10 ml syringe one inch at a time. Immediately after injection, the layers were clamped down and held for 20 seconds to ensure no cracks while acrylic bonds were forming and repeated four times.

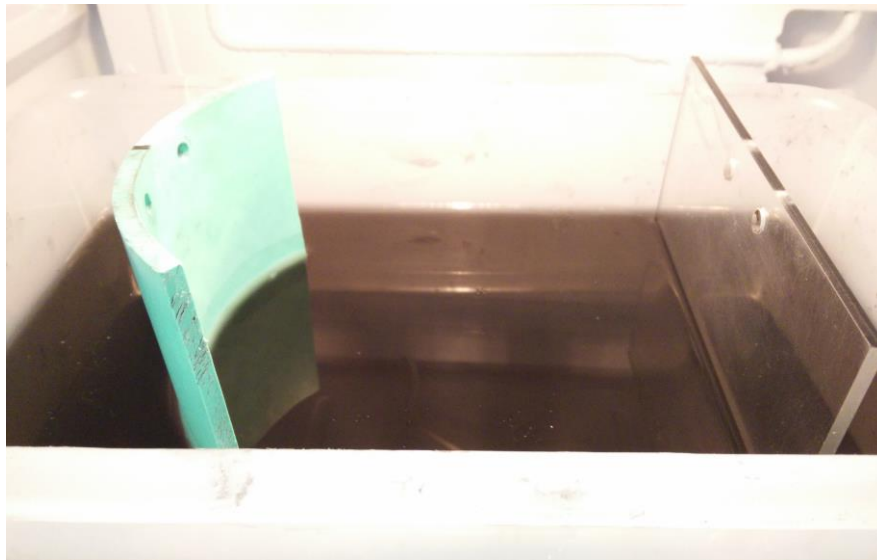
The outside faces were cut with an overlap on the inside giving space to attach the spacer [Figure 13]. Dichloromethane is applied to the outside frame to attach the space frame. The Mylar sheet was sandwiched between the outside frame and the inside frame. This component applies even pressure on the Mylar membrane and making it flush on the inside of the box to prevent boundary effects for shear wave propagation. This process was repeated four times for each side of the box.



*Figure 12 - Outside frame with extra material attached to a spacer frame. The space frame fits snugly in the inside frame of the box*

The inner frames were then fitted to the base and dichloromethane was applied. The divots in each place should fit into the other plate and create a tight seal. At this time, silicone caulk was applied to the inner edges using a fine tipped caulking tool and left overnight to cure.

Indents were made at the top of the box designed to be a place holder for the VTS-100 shaker plates [Figure 14]. The plates can be positioned parallel to each other and as deep in the gel as possible. As you can see from figure 14, the plates are high and above the standard plastic box, but with the newly manufactured box, the plates can be positioned deeper. By embedding the plates deep into the gelatin, the transducer will have enough flexibility to get as close to the source of shockwave.

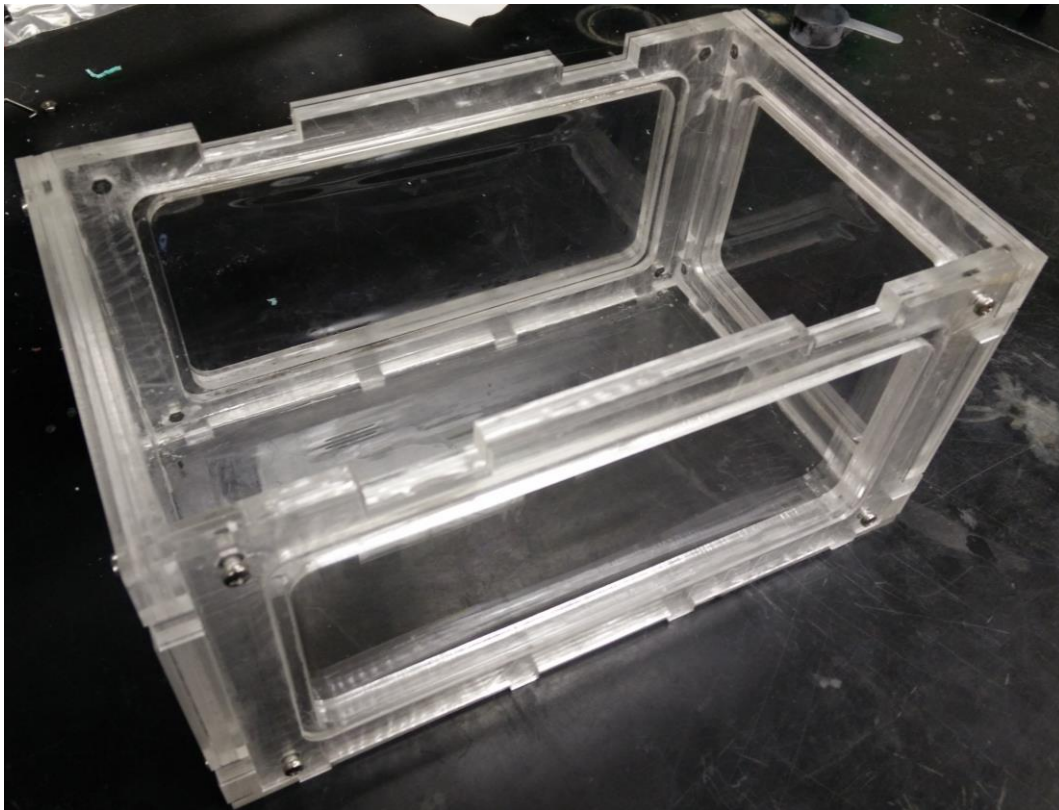


*Figure 13 - Shaker plates embedded in the gel. The indents in the windowed box will allow for easier placement of the plates.*

Mylar windows were easily cut into rectangles that are sufficiently wide enough to cover the screw holes in the frame. A hole can then be punched in the Mylar to match the holes in the frame and the M4 screws can be fed into it. We then maintained tension in the Mylar film by evenly tightening each screw causing the outside frame and spacer to press

the film into the box. To assist in the tightening process, clamps were used to ensure equal tension in the film. The Mylar sheet can also benefit from a coat of methylene chloride to help hold it in place and create more friction. After the frame is completely tightened into the box, the Mylar film was subjected to an industrial heat gun to treat the Mylar film.

The Mylar film should not have any wrinkles or defects. The process was repeated three more times for each remaining face [Figure 15]. The resultant box should now be able to hold water and limit the amount of artifacts when scanning with the ultrasound transducer.



*Figure 14 - Mylar windowed box for ultrasound scanning on different planes.*

### III. Programming

Targets in space can be defined relative to the base of the robot (Wobj) and will have defined orientation and configurations for the robot which Robotstudio will convert to RAPID code. Configurations of targets with linear commands should be the same. For example, a configuration of [1,0,0,0] should be used for all targets that lie on the top of the box container. A configuration of [-0.5,0.5,0.5,-0.5] can be used for targets defined on the right window of the box. Matching configurations will ensure that the robot only changes orientation for different faces of the box. After the targets were defined, they can be placed into a path which will create RAPID code inside the main() loop. Only targets defined in paths will be converted to RAPID.

Before synchronization with the system workspace, a tool for the ultrasound transducer was defined. A dummy tool can be created by defining the weight in kg and the center of gravity of the tool. For this study, the tool that was defined was the ultrasound transducer in both the short holder and the long holder. For the tool, the origin is where the tip of the transducer lies in space relative to the robot arm attachment when it is cradled in the holders. Table 3 shows the x, y, and z coordinates of both the short and long holders relative to the base of the robot. Once a tool is created, Robotstudio will create a RAPID library and RAPID tool name that can be used in the editor. Moreover, the tool can be attached to the end of the robot arm virtually. However, for the different holders, new targets must be defined and configured to match the new origin point of the transducer.

Table 4 – Tool definition coordinates for the ultrasound transducer in different attachments

ATTACHMENT	X COORDINATE (MM)	Y COORDINATE (MM)	Z COORDINATE (MM)	WEIGHT (KG)
SHORT HOLDER	-61	0	108	0.582
LONG HOLDER	53.23	71.04	97.04	0.556

Rapid code should now have defined targets, tools, configuration, orientation, and joint data. All of this data should be saved in a .mod file called module 1. To move the robot arm linearly, the desired coordinates can be uploaded to the robot controller. To rotate the robot arm, quaternions first need to be calculated and then uploaded as new RAPID code. For simple rotations, configurations and quaternions can be predefined in RobotStudio and used interchangeably in the code. The RAPID code may be synchronized with the virtual station to create a simulation before transferring the module 1 file from the computer to the Robot Controller (120-502457). After updating the .mod file, the ABB IRB-120 can be controlled by constant back-and-forth between the robot and the PC.

To trigger and sync the robot movement to the ultrasound transducer, an executable must be created to interface with Matlab. The ultrasound scanner will send a trigger pulse that will run the executable and update the .mod file. The executable was created using Visual Studio C# and Robotstudio PC SDK (Software Development Kit). The .cs file [Appendix A] can then be turned into a .bat file for access through the Fedora Linux terminal and the Oracle Virtual Machine VirtualBox Manager Windows 7 boot. Now, the Linux terminal connected to the ultrasound transducer can communicate with the Linux terminal connected to the robot arm. A 3D array of coordinates can be created in Matlab and the robot can iterate through each of these steps to create a 3D model.

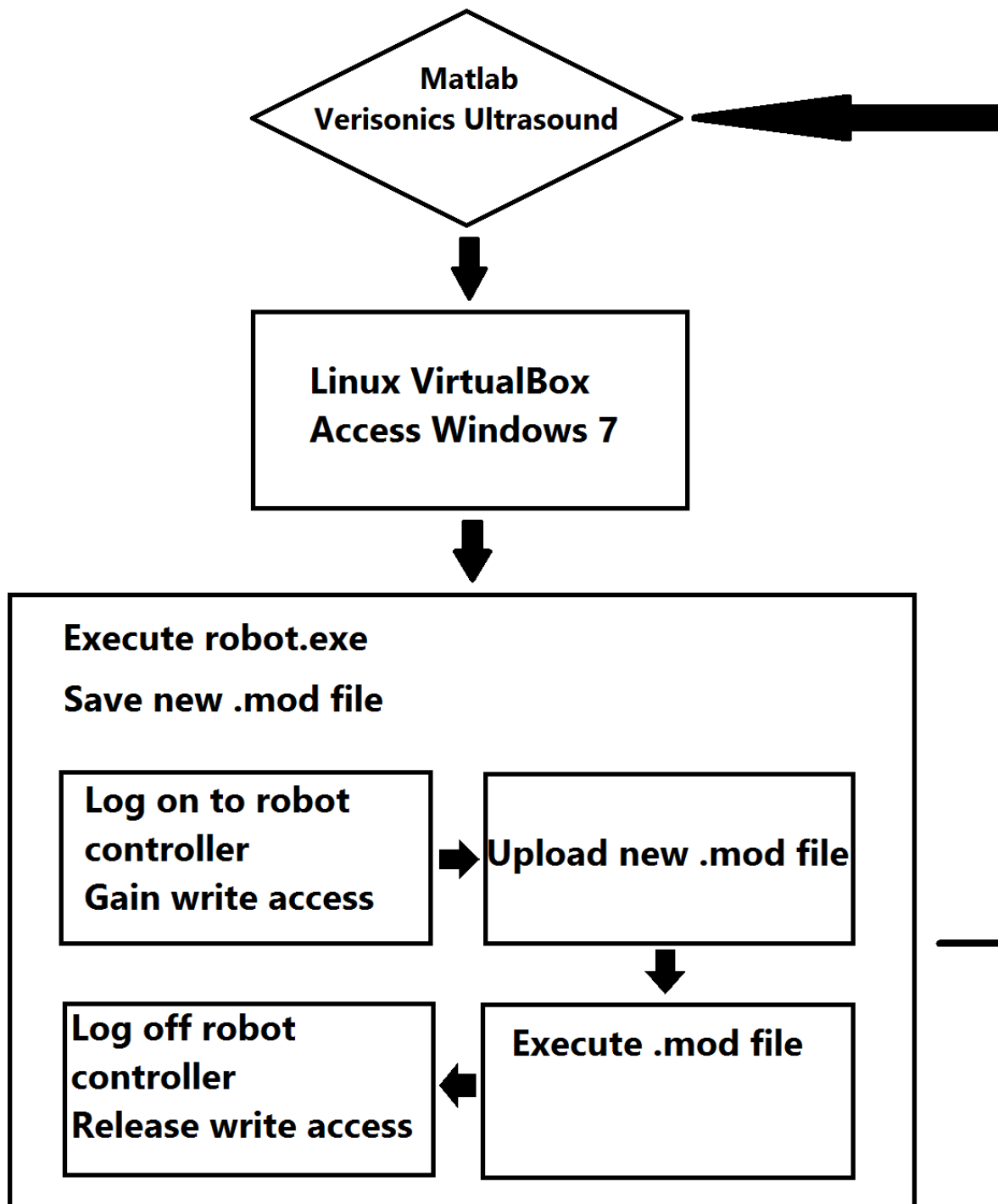
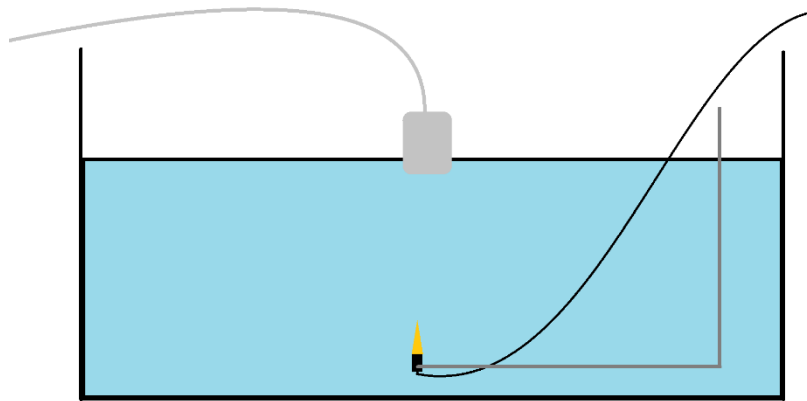


Figure 15- Flowchart for programming sequence.

#### IV. Dataset Registration of Angle Measurements

##### A. 3 Dimensional Hydrophone Scanning

A hydrophone [Onda Corporation, Sunnyvale, CA] was used for the data and angle registration. The hydrophone was connected to a pre-amplifier and was then connected to the Pico oscilloscope [Pico Technology, United Kingdom]. The hydrophone can then be placed safely inside a water tank in the orientation according to Figure 16. The ultrasound transducer can now be moved using the robotic programming to find the surface of the water. The surface of the water becomes the baseline for the remainder of the scan.



*Figure 16- Drawing of the hydrophone and water tank set up*

At each data point, the ultrasound transducer was triggered to produce a 5MHz pulse and that same trigger was used to trigger the recording of the hydrophone. The hydrophone will take the data recorded for 5 ms and plot the intensity according to the 3 dimensional array. The Matlab code should then create three different images of the 5MHz pulse in X-Y, Y-Z, and Z-X. The resultant images can be interpolated into a cohesive 3 dimensional image of the 5MHz pulse.

The intensity of the beam was recorded in a 15 x 15 x 15 array of intensities. Matlab code was generated to find the max intensities in each z plane and record the location of the max. A linear regression was then made to plot the angle of the transducer.

## B. Planar Wave Scanning

A second experiment was conducted to get better angle registration of the transducer. For this part, only the ultrasound transducer was used. A planar wave is generated from the ultrasound transducer so it propagates through water. The plane wave reflects off of the bottom surface of the water tank and the ultrasound record the echo.

## V. Results

### A. Hydrophone Beam Plotting Angle Displacement

The angle of the surface of the table was zeroed and the angle of the transducer relative to the table is about 0.02 degrees according to the Wixley Digital Angle Gauge.

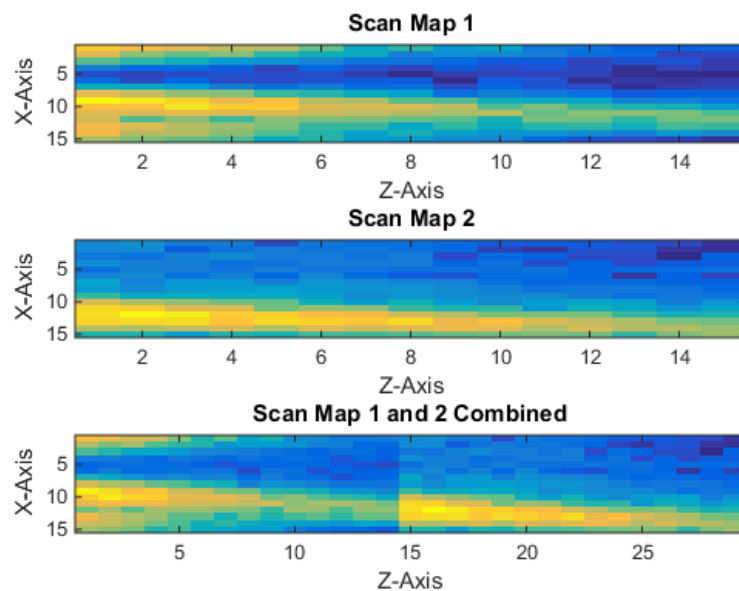
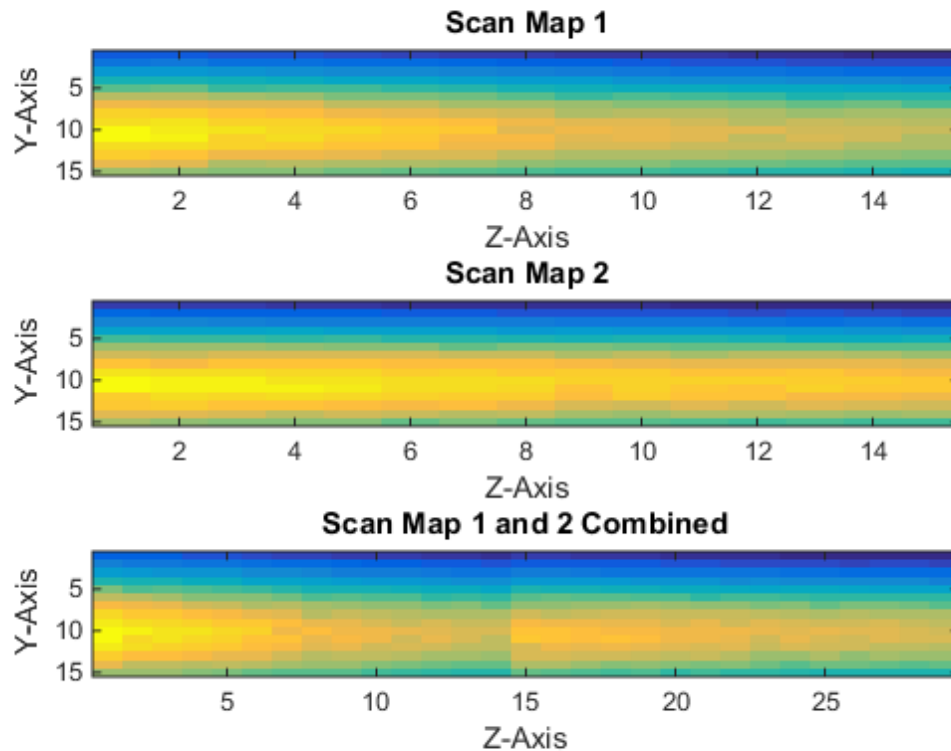


Figure 17 – intensity beam of the ultrasound plotted



The figure above shows the results of the beam plotting. The first scan was taken a day before the second scan. Both experiments used a 15x15x15 array but they overlap by one pixel. Therefore the final graph above is a 15x15x29 array rather than a 15x15x30. The beam in the above figure visually looks slanted.



*Figure 18 - plot of Y-Z beam plotting*

Figure 18 shows the beam plot for each individual scan and the combined scan in the Y-Z plane. Observationally in each individual scan, there seems to be a slight slope in the intensity map of the beam. However, the combined intensity graphed appears more flat suggesting no angle displacement from the Z-axis.

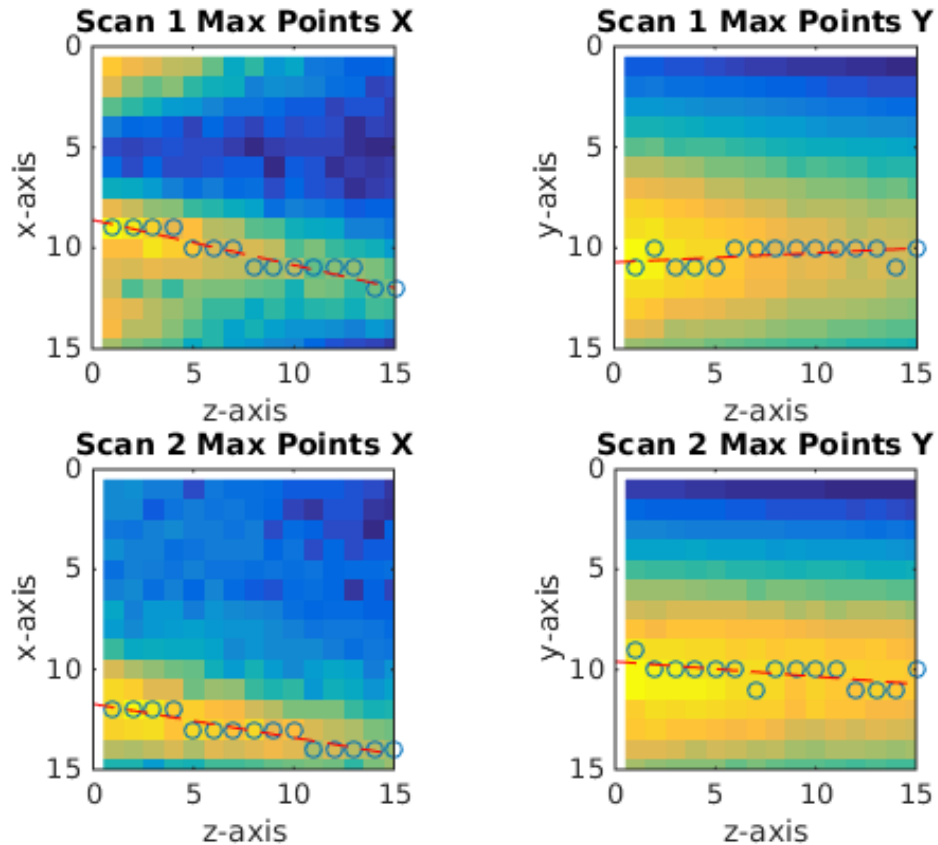


Figure 19 - beam plot of individual scans separated

The data from the scan was first analyzed separately. Visually, we can note that there appears to be a noticeable slope in each graph. Many of the data points in the Y-Z plane were centered around 10 units whereas in the X-Z plane, there is a steady decrease from 12 to 9 units.

Linear regressions for XZ plane:

$$y = 0.225x + 8.6$$

$$y = 0.1679x + 11.7238$$

Linear regressions for the YZ plane:

$$y = -0.0464x + 10.704$$

$$y = 0.075x + 9.6$$

Individually, each graph has a noticeable slope which indicates that the transducer is at an angle to the hydrophone. The displacement angle in the x-axis is 12.6804 and 9.5287 degrees and the displacement in y is -2.6583 and 4.2892 degrees. These results differ significantly from the initial reading of the digital angle gauge.

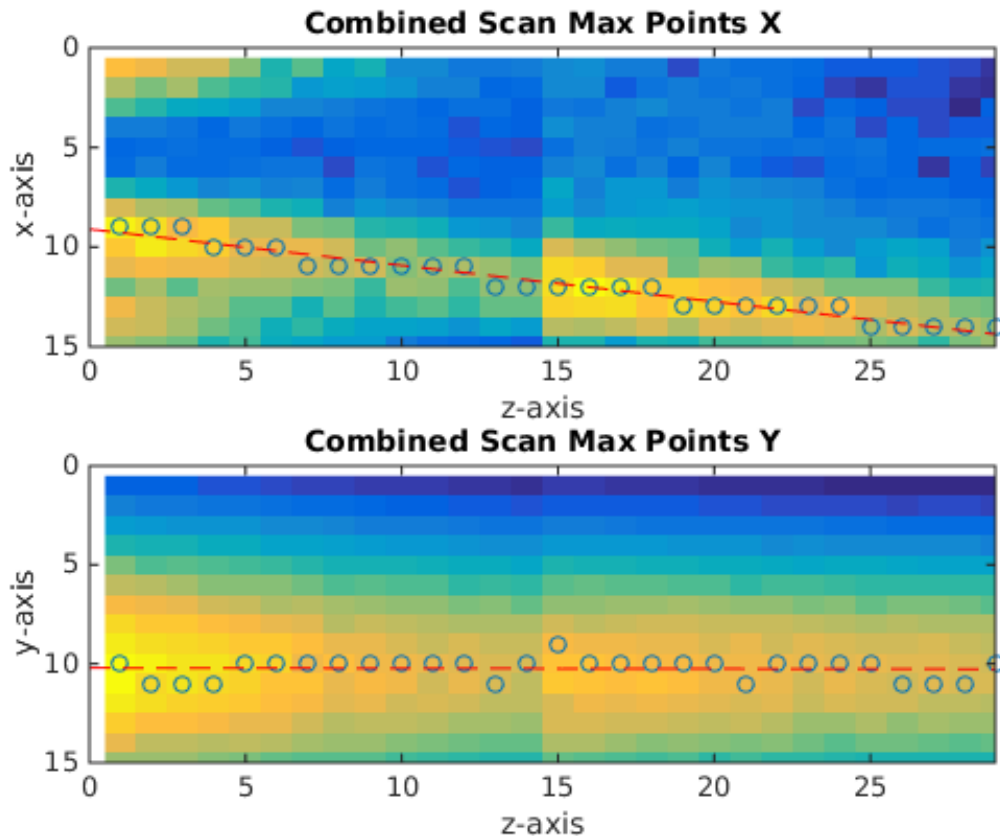


Figure 20 - linear regression superimposed on beam plots of both scans combined. There is a slope on the XZ plane but a smaller slope on the YZ axis.

The maximum intensity of the beam was plotted and super imposed on the beam plot and a linear regression was made to fit the data. The linear regression for the X-Z plane beam is  $y = .1818x + 9.1010$  and the linear regression for the Y-Z plane is  $y = 0.002x + 10.2118$ .

There is a slight slope for the x axis beam and a slope that is almost zero in the y. The displacement angle is calculated by taking the inverse tangent of two vectors. In this case, the vectors are the linear regression line and the z – axis. The angle of the beam in the XZ plane is 10.3024 degrees and 0.1129 degrees in the YZ plane. Though, the concatenated data has a decreased angle with respect to the z – axis, it still does not comply with our initial reading of 0.2 degree difference. What is most likely happening in the above scan is that we are seeing the secondary lobe of a Sinc function rather than the targeted beam.

### B. Planar Wave Angle Displacement

$$\lambda = \frac{\text{speed of sound in water}}{\text{frequency}} = \frac{1484 \text{ m/s}}{5200000 \text{ Hz}} = 2.85 \times 10^{-4} \text{ meters}$$

The Ultrasound Transducer emits a 5.2 MHz planar wave with 8 points per wavelength. Therefore, there is 8 points per every  $2.85 \times 10^{-4}$  meters or 1 point is  $3.57 \times 10^{-5}$  meters. The element spacing inside the transducer is 0.298 mm for every point and the width of the transducer is 38.293 mm. Every point on the y-axis represents  $3.57 \times 10^{-2}$  mm and every point on the x-axis represent 0.298 mm in the figure below.

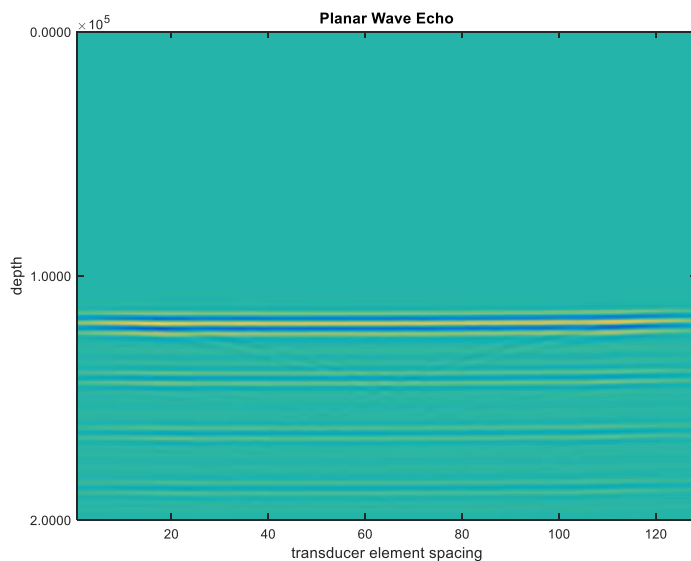


Figure 21 - planar wave echo produced by ultrasound transducer

The maximum intensities of the planar wave was plotted and superimposed on figure 21 and a linear regression was fitted to the data points. The line does show a slight curve indicating that there is offset in the transducer positioning. The equation of the line is calculated in Matlab to be  $y = -0.0001x + 1.2072$ . The slope of the line is very small and indicates that the angle should not be more than 1 degree. Taking the element spacing and the points per wavelength into account, the resultant angle compared to the X-axis is calculated to be  $-5.9045 \times 10^{-9}$  degrees. The angle calculated is much smaller than the angle recorded from the digital angle gauge, indicating that the planar wave method is more accurate and that we can relate ultrasound data at two different points to one another. We can now use the robot arm for construction of 3D maps and eventually 7D.

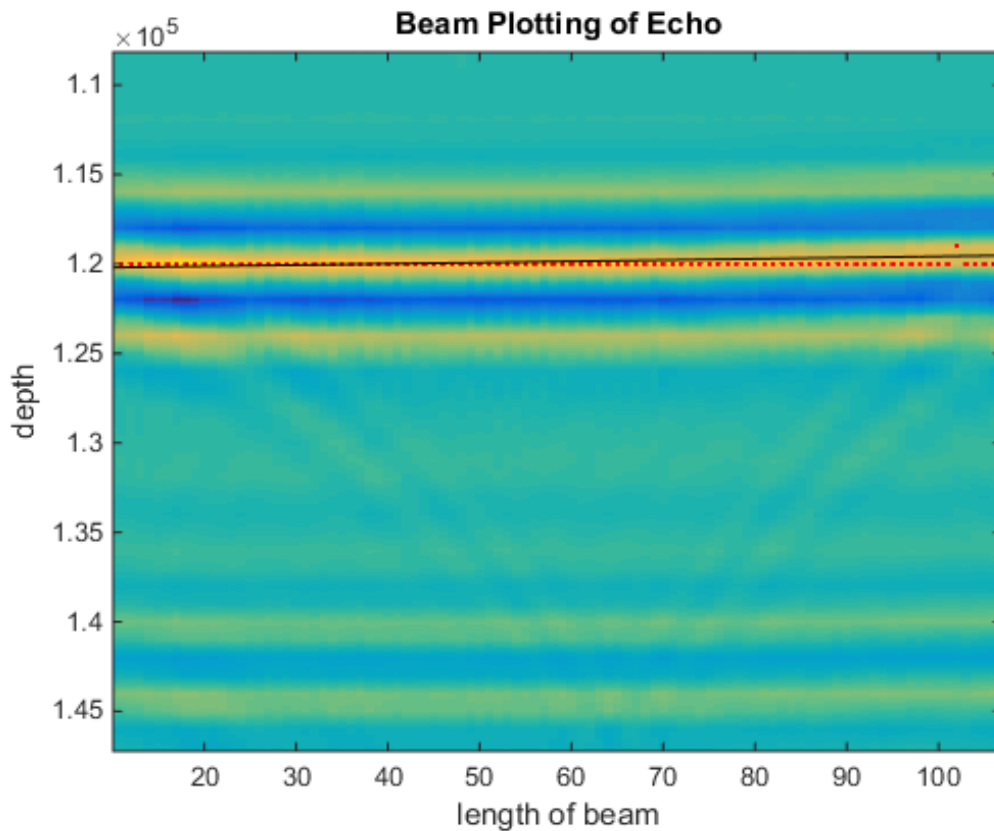


Figure 22 - Intensities plotted with a linear regression

## VI. Conclusion

In this study, we created a method to accurately position the ultrasound transducer using a hydrophone and an ABB IRB 120 Robot. The first objective was creating a method of attaching the ultrasound transducer to the robot arm in the same orientation and positioning every time. Low repeatability induces high error and less accurate models. We formed a Mylar box that allows the robot to scan on different planes without hindrance of artefacts. Various coding was completed that allows the robot to be triggered by the ultrasound transducer and the registration of 3 dimensional data sets calibrates the angle of the transducer. From the scan with the hydrophone we saw a .165 degree displacement in the y direction incident with the hydrophone and a 10 degree displacement in the x. Because the x was so large, we determined that it was a side lobe cause by a sinc function of the ultrasound and conducted another test for more assurance.

Various improvements can be made to the manufacturing processes. The Aluminum mold should be degassed before injection into the mold to prevent air bubbles. Although the air bubbles did not interfere with the positioning of the transducer, it does lower aesthetic quality. There should be more capture nut screws in the Mylar box to keep constant tension from all sides of the Mylar sheet. Wrinkles in the Mylar will reflect shear waves. Additionally, the handshakes between the ultrasound machine and the robot has a 5 second delay at each data point. This 5 second delay quickly accumulates to create longer scans that regulate the speed of each successive 3D scan. However, these processes will be improved with further research and studies.

The techniques developed in this study can be used to create 7 dimensional maps by conducting the 3D scans on different planes. Knowing the angle of incidence of the transducer is crucial to the production of accurate models and data sets. Without having accurate positioning, raw ultrasound data at one point cannot be related to the data at another point. Therefore, tissue displacement is not accurate in the time dimension. For the continuation of this study, the Mylar box will be used to scan different angles of the hydrophone and, eventually, brain tissue. By creating 7D data sets of ultrasound data, we can predict how brain tissue moves in a TBI leading to better prevention and treatment.

## VII. References

- [1] Centers for Disease Control and Prevention. (2006). Get the Stats on Traumatic Brain Injury in the United States. Retrieved from [http://www.cdc.gov/traumaticbraininjury/pdf/BlueBook\\_factsheet-a.pdf](http://www.cdc.gov/traumaticbraininjury/pdf/BlueBook_factsheet-a.pdf)
- [2] Thurman, David J., et al. "Traumatic brain injury in the United States: a public health perspective." *The Journal of head trauma rehabilitation* 14.6 (1999): 602-615.
- [3] Bigler, E. D. (2009). Traumatic brain injury. *Textbook of Alzheimer Disease and Other Dementias*, 229-246.
- [4] Catheline, S., Gennisson, J. L., Tanter, M., & Fink, M. (2003). Observation of shock transverse waves in elastic media. *Physical review letters*, 91(16), 164301.
- [5] Pinton, G., Gennisson, J. L., Tanter, M., & Coulouvrat, F. (2014). Adaptive motion estimation of shear shock waves in soft solids and tissue with ultrasound. *Ultrasonics, Ferroelectrics, and Frequency Control, IEEE Transactions on*, 61(9), 1489-1503.
- [6] Woo, J. (2002). A short history of the development of ultrasound in obstetrics and gynecology. See <http://www.ob-ultrasound.net/history1.html> (last checked 14 May 2011).
- [7] Chen, Y., & Dong, F. (2013). Robot machining: recent development and future research issues. *The International Journal of Advanced Manufacturing Technology*, 66(9-12), 1489-1497.
- [8] ABB Robotics. [www.abb.com/robotics](http://www.abb.com/robotics). Accessed 28 Feb 2012
- [9] ABB Robotics. (2014). ABB's smallest robot – for flexible and compact production. see [http://www09.abb.com/global/scot/scot241.nsf/veritydisplay/3bd625bab3c7cae1c1257a0800495fac/\\$file/ROB0149EN\\_D\\_LR.pdf](http://www09.abb.com/global/scot/scot241.nsf/veritydisplay/3bd625bab3c7cae1c1257a0800495fac/$file/ROB0149EN_D_LR.pdf)



[10] B. K. P. Horn. Some notes on unit quaternions and rotation.

<http://people.csail.mit.edu/bkph/articles/Quaternions.pdf>, 2001. 77, 80

[11] World Robotics. (2010). Worldwide operational stock of industrial robots [data].

<http://blog.robotiq.com/bid/52886/Industrial-robots-5-most-popular-applications>

[12] Hammond Manufacturing 1550p Enclosure. (2015). See

<http://www.hammondmfg.com/pdf/1550P.pdf>

## VIII. Appendix

## A. PC SDK - RAPID Code

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using ABB.Robotics;
using ABB.Robotics.Controllers;
using ABB.Robotics.Controllers.Discovery;
using ABB.Robotics.Controllers.RapidDomain;

namespace Robot_Communcation_v1_0
{
    class Program
    {
        static void Main(string[] args)
        {
            NetworkScanner scanner = new NetworkScanner();
            scanner.Scan();
//            Console.WriteLine("check 1");
//            Console.ReadKey();

            ControllerInfoCollection controllers = scanner.Controllers;
//            Console.WriteLine("check 2");
//            Console.ReadKey();

            foreach (ControllerInfo controllerinfo in controllers)
            {
                Console.WriteLine("IP Address:" + controllerinfo.IPAddress.ToString()+"\n");
                Console.WriteLine("System Name:" + controllerinfo.SystemName + "\n");
                Console.WriteLine("Controller Name:" + controllerinfo.ControllerName + "\n");
            }
            try
            {
                Console.WriteLine("Connected to IP:" +
scanner.Controllers[2].IPAddress.ToString());
            }
            catch(System.IndexOutOfRangeException ex)
            {
                Console.WriteLine("No controller Available" + ex.Message);
                Console.ReadKey();
            }
        }
    }
}

```

```

//      Console.WriteLine("check 3");
//      Console.ReadKey();

Controller controller = new Controller(scanner.Controllers[2].SystemId);
controller.Logon(UserInfo.DefaultUser);
//      Console.WriteLine("check 4");
//      Console.ReadKey();

string home = controller.GetEnvironmentVariable("HOME");
string remoteDir = controller.FileSystem.RemoteDirectory;
string localDir = controller.FileSystem.LocalDirectory;

Console.WriteLine(remoteDir);
Console.WriteLine(localDir);

try
{
    using(Mastership m = Mastership.Request(controller.Rapid))
    {
        controller.FileSystem.LocalDirectory = localDir;
        controller.FileSystem.RemoteDirectory = remoteDir;
        controller.FileSystem.PutFile("Module1.mod", "Module1.mod", true);
        //controller.FileSystem.PutFile("CalibData.mod", "CalibData.mod", true);
        //controller.FileSystem.PutFile("Rectangle.pgf", "Rectangle.pgf", true);
        ABB.Robotics.Controllers.RapidDomain.Task tMain =
controller.Rapid.GetTask("T_ROB1");
        //tMain.LoadProgramFromFile("test3.pgf", RapidLoadMode.Replace);
        //tMain.LoadModuleFromFile("Module1.mod", RapidLoadMode.Add);
        tMain.LoadModuleFromFile("Module1.mod", RapidLoadMode.Replace);
        //tMain.LoadModuleFromFile("CalibData.mod", RapidLoadMode.Replace);
        //tMain.LoadProgramFromFile("test3.pgf", RapidLoadMode.Replace);
        tMain.ResetProgramPointer();
    }
}
catch(Exception ex)
{
    Console.WriteLine("Failure to upload: " + ex.Message);
    Console.ReadKey();
}

//      controller.FileSystem.CopyFile(localDir + "test3.pgf", remoteDir + "test3.pgf", true);
//      controller.FileSystem.CopyDirectory(localDir, remoteDir, true);

//      Console.WriteLine("check 4.5");

```

```

//      Console.ReadKey();
try
{
    if (controller.OperatingMode == ControllerOperatingMode.Auto)
    {
        using (Mastership m = Mastership.Request(controller.Rapid))
        {

            controller.Rapid.Start();
        }
    }
    else
    {
        Console.WriteLine("Automatic Mode is required to start execution");
    }
}
catch (System.InvalidOperationException ex)
{
    Console.WriteLine("Mastership is held by another client. " + ex.Message);
    Console.ReadKey();
}
catch (System.Exception ex)
{
    Console.WriteLine("Unexpected error occurred: " + ex.Message);
    Console.ReadKey();
}
//      Console.WriteLine("check 5");
//      Console.ReadKey();
}

public static bool uploadsucces { get; set; }
}

```

## B. Linux Communication with module file

```
CD C:\Users\lab
```

```
START C:\Users\lab\tmp.exe
```

```
VBoxManage guestcontrol "Windows7" execute --image
```

```
"C:\Users\lab\runrobot.bat" --username lab --password ***** --verbose
```

```
--wait-exit
```

### C. Matlab code for angle calibration using a hydrophone

```

load('/home/lab/Documents/master_robot_transducer_calibration3d_c_workspace.m
at')
scanmatI=zeros(size(scanmat,2),size(scanmat,3),size(scanmat,4));

for ii=1:length(xvec)
    for jj=1:length(yvec)
        for kk=1:length(zvec)

            vec=scanmat(:,ii,jj,kk);
            fvec=fft(vec);
            fvec(1:length(filtvec))=fvec(1:length(filtvec)).*filtvec;
            fvec(length(filtvec)+1:end)=0;

            vec2=ifft(fvec,'symmetric');
            [val idx]=max(abs(hilbert(vec2)));
            ppw=round(1/(16e-9*5e6));
            scanmatI(ii,jj,kk)=sum(vec2(idx-ppw:idx+ppw).^2);
        end
    end
end

for ii=1:length(xvec)
    for jj=1:length(yvec)
        for kk=1:length(zvec)

            vec=scanmat2(:,ii,jj,kk);
            fvec=fft(vec);
            fvec(1:length(filtvec))=fvec(1:length(filtvec)).*filtvec;
            fvec(length(filtvec)+1:end)=0;

            vec2=ifft(fvec,'symmetric');
            [val idx]=max(abs(hilbert(vec2)));
            ppw=round(1/(16e-9*5e6));
            scanmat2I(ii,jj,kk)=sum(vec2(idx-ppw:idx+ppw).^2);
        end
    end
end

scanmatIminus=scanmatI(:,:,2:15);
scanmat3I = cat(3,scanmatIminus,scanmat2I);

figure
subplot(3,1,1);
imagesc(squeeze(scanmatI(round(end/2),:,:)))
xlabel('Z-Axis')
ylabel('X-Axis')
title('Scan Map 1')
subplot(3,1,2);

```

```

imagesc(squeeze(scanmat2I(round(end/2),:,:)))
xlabel('Z-Axis')
ylabel('X-Axis')
title('Scan Map 2')
subplot(3,1,3);
imagesc(squeeze(scanmat3I(round(end/2),:,:)))
xlabel('Z-Axis')
ylabel('X-Axis')
title('Scan Map 1 and 2 Combined')

%[val idx idy idz]=max3d(scanmatI)
%idx=3;
%% getting max/location
scanmatout = scanmatI;
scanmatout(1,:,:) = 0; % getting rid of outliers
A = zeros(1,15);
B = 1:15;
for i = 1:length(scanmatout)
    A(i) = max(max(scanmatout(:,:,i)));
    [row(i),col(i)] = find(scanmatout(:,:,i)==A(i));
end
%imagesc(scanmatI(:,:,,:));

AA = zeros(1,15);
BB = 1:15;
for i = 1:length(scanmat2I)
    AA(i) = max(max(scanmat2I(:,:,i)));
    [row2(i),col2(i)] = find(scanmat2I(:,:,i)==AA(i));
end

AAA = zeros(1,29);
BBB = 1:29;
for i = 1:length(scanmat3I)
    AAA(i) = max(max(scanmat3I(:,:,i)));
    [row3(i),col3(i)] = find(scanmat3I(:,:,i)==AAA(i));
end
figure
subplot(3,1,1);
plot3(row,col,B)
axis ([0 15 0 15 0 15])
xlabel('X-Axis');
ylabel('Y-Axis');
title('Scan Map 1');

subplot(3,1,2);
plot3(row2,col2,BB)
axis ([0 15 0 15 0 15])
xlabel('X-Axis');
ylabel('Y-Axis');
title('Scan Map 2');

subplot(3,1,3);
plot3(row3,col3,BBB)
axis ([0 15 0 15 0 15])
xlabel('X-Axis');
ylabel('Y-Axis');

```

```

title('Scan Map Combined');
%% plotting x vs z and y vs z

figure
subplot(2,2,1);
imagesc(squeeze(scanmatI(:,round(end/2),:)))
hold on
plot(B,row,'o');
axis ([0 15 0 15]);
axis square
p1 = polyfit(B,row,1);
xline1 = linspace(0,15,200);
yline1 = p1(1)*xline1+p1(2);
hold on
plot (xline1,yline1,'--r')
ylabel('x-axis')
xlabel('z-axis')
title('Scan 1 Max Points X')
hold off

subplot(2,2,2);
imagesc(squeeze(scanmatI(round(end/2),:,:)))
hold on
plot(B,col,'o');
axis ([0 15 0 15]);
axis square
p2 = polyfit(B,col,1);
xline2 = linspace(0,15,200);
yline2 = p2(1)*xline2+p2(2);
hold on
plot (xline2,yline2,'--r')
ylabel('y-axis')
xlabel('z-axis')
title('Scan 1 Max Points Y')
hold off

subplot(2,2,3);
imagesc(squeeze(scanmat2I(:,round(end/2),:)))
hold on
plot(B,row2,'o');
axis ([0 15 0 15]);
axis square
pp1 = polyfit(BB,row2,1);
xxline1 = linspace(0,15,200);
yyline1 = pp1(1)*xxline1+pp1(2);
hold on
plot (xxline1,yyline1,'--r')
ylabel('x-axis')
xlabel('z-axis')
title('Scan 2 Max Points X')
hold off

subplot(2,2,4);
imagesc(squeeze(scanmat2I(round(end/2),:,:)))
hold on
plot(BB,col2,'o');

```



```

axis ([0 15 0 15]);
axis square
pp2 = polyfit(BB,col2,1);
xxline2 = linspace(0,15,200);
yyline2 = pp2(1)*xxline2+pp2(2);
hold on
plot (xxline2,yyline2,'--r')
ylabel('y-axis')
xlabel('z-axis')
title('Scan 2 Max Points Y')
hold off

figure
subplot(2,1,1);
imagesc(squeeze(scanmat3I(:,round(end/2),:)))
hold on
plot(BBB,row3,'o');
axis ([0 29 0 15]);
ppp1 = polyfit(BBB,row3,1);
xxxline1 = linspace(0,29,200);
yyyline1 = ppp1(1)*xxxline1+ppp1(2);
hold on
plot (xxxline1,yyyline1,'--r')
ylabel('x-axis')
xlabel('z-axis')
title('Combined Scan Max Points X')
hold off

subplot(2,1,2);
imagesc(squeeze(scanmat3I(round(end/2),:,:)))
hold on
plot(BBB,col3,'o');
axis ([0 29 0 15]);
ppp2 = polyfit(BBB,col3,1);
xxxline2 = linspace(0,29,200);
yyyline2 = ppp2(1)*xxxline2+ppp2(2);
hold on
plot (xxxline2,yyyline2,'--r')
ylabel('y-axis')
xlabel('z-axis')
title('Combined Scan Max Points Y')
hold off
%% Obtaining Angle Displacement

xanglerad = atan2(yline1(200)-yline1(1),xline1(200)-xline1(1));
xangledeg = atan2(yline1(200)-yline1(1),xline1(200)-xline1(1))*180/pi

yanglerad = atan2(yline2(200)-yline2(1),xline2(200)-xline2(1));
yangledeg = atan2(yline2(200)-yline2(1),xline2(200)-xline2(1))*180/pi

xangle2rad = atan2(yyline1(200)-yyline1(1),xxline1(200)-xxline1(1));
xangle2deg = atan2(yyline1(200)-yyline1(1),xxline1(200)-xxline1(1))*180/pi

yangle2rad = atan2(yyline2(200)-yyline2(1),xxline2(200)-xxline2(1));
yangle2deg = atan2(yyline2(200)-yyline2(1),xxline2(200)-xxline2(1))*180/pi

```

```
xangle3rad = atan2(yyyline1(200)-yyyline1(1),xxxline1(200)-xxxline1(1));  
xangle3deg = atan2(yyyline1(200)-yyyline1(1),xxxline1(200)-  
xxxline1(1))*180/pi
```

```
yangle3rad = atan2(yyyline2(200)-yyyline2(1),xxxline2(200)-xxxline2(1));  
yangle3deg = atan2(yyyline2(200)-yyyline2(1),xxxline2(200)-  
xxxline2(1))*180/pi
```

## Appendix D – Matlab Coding for Planar Wave Registration

```

figure
M = cell2mat(RcvData);
space = .298;
depth = (1484/5200000)/8;
spacecalibration = space/256;
depthcalibration = depth/2048;
imagesc([space space*256],[depth depth*2048],M)
A = zeros(1,128);
B = 1:128;
row = zeros(1,128);
N = double(M);
x = 1:1:2048;
xq = 1:0.001:2048;
%Nm = linspace(1,length(N),2*length(N));
NN = interp1(x,N,xq);
NN(1:200000,:) = 0;
Nq = NN(1300000:1500000,1:128);
for i = 1:length(A)
    A(i) = max(max(NN(:,i)));
    row(i) = find(NN(:,i) == A(i));
end
AA = zeros(1,128);
for j = 1:length(AA)
    AA(j) = max(max(Nq(:,j)));
    row2(j) = find(Nq(:,j) == AA(j));
end
%% plotting linear regressions on to imagesc

figure
imagesc(NN(1300000:1500000,1:128));
title('Planar Wave Echo')
xlabel('transducer element spacing')
ylabel('depth')
hold on
plot(B,row2,'.r');
p1 = polyfit(B,row2,1);
xline1 = linspace(0,128,400);
yline1 = p1(1)*xline1+p1(2);
hold on
plot(xline1,yline1,'-k')
ylabel('depth')
xlabel('length of beam')
title('Beam Plotting of Echo')
hold off
%% angle calculation with depth and spacing calibration
yline = yline1/10e5;
xline = xline1;
yy = yline(400) - yline(1);
xx = xline1(400) - xline1(1);
thetarad = atan2(yy*depthcalibration,xx*spacecalibration);
thetadeg = thetarad*(180/pi)

```